REMARKS

Status of the Claims

Claims 21-40 are currently present in the Application, and claims 21, 28, and 34 are independent claims. Claims 1-20 have been cancelled, and claims 21-40 have been added in this Amendment.

Drawings

Applicants note with appreciation the acceptance, by the Examiner, of Applicants' formal drawings that were submitted with the Application.

Claim Objections Under 35 U.S.C. § 112

The Office Action objected to the title of Applicants' invention as being non-descriptive. Applicants have amended the title of the invention. The new title is clearly indicative of the invention to which the new claims are directed. Accordingly, Applicants respectfully request that the Examiner withdraw the objection to Applicants' title.

Claim Rejections - 35 U.S.C. § 101

Applicants' original computer program product claims (14-20) were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter as including "signals for downloading via the Internet or other network." Applicants respectfully direct the Examiner's attention to Applicants' amendment to the specification, particularly the amendment to the first paragraph of page 29 where Applicants remove the language that computer program products include such signals downloaded from the Internet or other computer network. Consequently, in light of Applicants' amendment, Applicants respectfully request that the Examiner withdraw the rejection of Applicants computer program product claims (now new claims 34-40).

Claim Rejections – Alleged Obviousness Under 35 U.S.C. § 103

Original claims 1-20 each stood rejected under 35 U.S.C. § 103 as being obvious and therefore unpatentable over U.S. Patent No. 3,800,291 to Cocke et al. (hereinafter

"Cocke") in view of various other references. In particular, claims 1, 6-7, 14, and 19-20 were rejected as being obvious, and therefore unpatentable, over Cocke in view of Applicants' background section. Claims 2, 3, 15, and 16 were rejected as being obvious, and therefore unpatentable, over Cocke in view of Applicants' background section in further view of U.S. Patent No. 4,868,738 to Kish et al. (hereinafter "Kish"). Finally, claims 4, 5, 8, 11-13, 17, and 18 were rejected as being obvious, and therefore unpatentable, over Cocke in view of Applicants' background section in further view of U.S. Patent No.4,446,517 to Katsura et al. (hereinafter "Katsura").

Applicants have canceled claims 1-20 in this amendment. Therefore, the rejections of claims 1-20 as being unpatentable under 35 U.S.C. § 103 are now moot. In the newly presented claims, Applicants have redirected and refocused the claimed invention to previously unclaimed subject matter regarding Applicants' disclosed method, information handling system, and computer program product of performing just-in-time (JIT) compilations. Below, Applicants set forth the support for the newly added claims in an effort to aid the Examiner in the examination of these newly added claims.

Claims 21, 28, and 34 claim a method, information handling system, and computer program product, respectively, that each include limitations of performing just-in-time (JIT) compilations by:

- initializing a mapping table, wherein the mapping table is used to track JIT compilations performed by the JIT compiler;
- initializing a special filesystem that is used for memory mapping JIT compiled code, wherein the special filesystem responds to requests received from an operating system by:
    - responding to a load request to load a page by writing one or more invalid opcodes to the page and by returning a first successful completion code to the operating system; and
    - responding to a write request to write one or more pages from memory to a nonvolatile storage device by returning a second

successful return completion code to the operating system without writing the one or more pages to the nonvolatile storage device;

- registering an error handler with the operating system, wherein the error handler is called by the operating system when the invalid opcode is encountered; and
- creating a special file using the special filesystem, wherein the special file is memory mapped to an address space, and wherein the address space is used to store JIT compiled code resulting from the JIT compiler compiling a plurality of methods.

Support for these claims is found in Figures 6, 8, and 9 and corresponding text from the specification which is found at page 17, line 15 to page 20, line 3 (Fig. 6), page 21, line 24 to page 23, line 26 (Fig. 8), and page 23, line 27 to page 26, line 24 (Fig. 9). Figure 6 deals with general steps taken to initialize the JIT compiler, Figure 8 deals with steps taken for the special filesystem to handle requests to write pages to nonvolatile storage, and Figure 9 deals with how the special filesystem loads a page after a page fault is encountered.

New claims 22, 29, and 35 are dependent claims that each add limitations of:

- detecting, by the operating system, an instruction that causes a page fault resulting from a branch to a non-loaded page of memory;
- requesting the special filesystem to handle the page fault;
- receiving, at the special filesystem, the request to handle the page fault, wherein the special filesystem responds by writing the one or more invalid opcodes to one or more first addresses within a first address range within the address space, the first address range corresponding to the non-loaded page, and the special filesystem returns the first successful completion code to the operating system;
- restarting, by the operating system, the instruction that caused the page fault, the restarting resulting in an invalid opcode error due to

the one or more invalid opcodes written by the special filesystem to the first addresses; and

- calling, by the operating system, the registered error handler in response to the invalid opcode error, wherein the error handler operates by:
  - loading a plurality of instructions in the first address range within the address space, the loading overwriting the invalid opcodes stored a the first addresses; and
  - restarting the instruction that caused the page fault, the restarting resulting in execution of the loaded instructions.

Support for new claims 22, 29, and 35 can be found in Figure 9 and corresponding specification description which is found on page 23, line 27 to page 26, line 24. These claims provide additional limitations for the interaction between the operating system memory manager, the special file system, and the registered error handler in loading a page when a page fault is detected.

New claims 23, 30 and 36 are dependent on claims 22, 29, and 35, respectively, and each further clarifies that the loading of the instructions by the error handler includes further limitations of:

- receiving a page address of the non-loaded page;
- identifying a method name from a plurality of method names stored in the mapping table, wherein the identified method name corresponds to the page address; and
- writing the plurality of instructions to the first address range, wherein the plurality of instructions correspond to a method corresponding to the identified method name, and wherein the writing overwrites the one or more invalid opcodes previously written to the one or more first addresses.

Support for newly added claims 23, 30, and 36 can be found in Figure 9, elements 960 to 998 which details the operation of the error handler in loading code (see also, specification page 25, line 19 to page 26, line 24 which details the operation of the error handler).

Newly added claims 24, 31, and 37 are dependent on claims 23, 30, and 36, respectively, and add further limitations regarding how the error handler determines whether to re-compile (reJIT) code or load interpreted bytecode instead. Each of these claims add further limitations of:

- determining whether the page fault occurred at the beginning of the method, wherein the writing of the plurality of instructions is performed by JIT compilation when the page fault does not occur at the beginning of the method;
- analyzing usage statistics corresponding to the method in response to the page fault occurring at the beginning of the method;
- in response to the analysis revealing a high usage of the method:
    - performing the JIT compilation, wherein the JIT compilation results in the writing of the plurality of instructions; and
- in response to the analysis revealing a lower usage of the method:
    - loading bytecode instructions into the first address range; and
    - removing the identified method name from the mapping table.

Support for newly added claims 24, 31, and 37 can be also found in Figure 9, elements 960 to 998 which details the operation of the error handler in loading code (see also, specification page 25, line 19 to page 26, line 24 which details the operation of the error handler).

New claims 26, 33, and 39 each depend on independent claims 21, 28, and 34, respectively, and each add limitations of:

- determining that additional address space is needed to store additional JIT compiled code, wherein the additional JIT compiled code corresponds to one or more additional methods, and wherein the mapping table lists each method currently stored in the special file along with each method's address range;
- analyzing usage statistics corresponding to the JIT code currently stored in the special file;
- identifying one or more methods corresponding to seldom used JIT code; and
- reclaiming address space from the special file by removing an entry corresponding to each of the identified seldom used methods from the mapping table.

Support for newly added claims 26, 33, and 39 can be found in Figures 7 and 8 and in the specification at page 20, line 4 to page 23, line 26. Each of these claims is directed towards memory reclamation.

Claims 27 and 40 are dependent on claims 26 and 39, respectively, and claim 33, in addition to the limitations set forth above, includes further limitations dealing with actions that are taken during memory reclamation when the special file used to store JIT compiled code is too small. These limitation include:

- creating a secondary special file using the special file system;
- memory mapping the secondary special file;
- receiving, from the operating system, a secondary address space resulting from the memory mapping; and
- initializing a secondary mapping table used to track the additional JIT compiled code stored in the secondary special file.

Support for claims 27, 40 and the additional limitations of claim 33 can be found in Figure 7 (elements 760 to 785) and the detailed description at page 20, line 28 to page 21, line 20.

<u>Conclusion</u>

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Joseph T. Van Leeuwen, Reg. No. 44,383/

Joseph T. Van Leeuwen, Reg. No. 44,383
Van Leeuwen & Van Leeuwen
Attorneys for Applicant
Telephone: (512) 301-6738
Facsimile: (512) 301-6742